AMENDMENTS TO THE CLAIMS

1. (currently amended) A method for scaling a digital picture to generate a scaled picture comprising the following steps:

(a) scaling a portion of the digital picture instead of the whole digital picture in a first direction;

(b) scaling part of the data produced in step (a) in a second direction; and

(c) for each different portion in sequence, repeating steps (a) and (b) to form the scaled picture.

2. (currently amended) The method of claim 1 wherein steps (a) and [[(c)]] (b) are performed by using an N-tap filter where N is ~~an even integer~~natural number a natural number.

3. (original) The method of claim 1 wherein the first direction is a horizontal direction, and the second direction is a vertical direction.

4. (original) The method of claim 1 wherein the first direction is a vertical direction, and the second direction is a horizontal direction.

5. (original) The method of claim 1 further comprising step (d): initializing a buffer used for storing the data produced by step (a).

6. (currently amended) The method of claim 5 wherein step (d) comprises ~~mirroring part of the digital picture scaled in step (a)~~ applying a mirror boundary condition to the buffer.

7. (currently amended) The method of claim 1 further comprising before scaling a last portion of the digital picture in the second direction, ~~mirroring~~ applying a mirror

boundary condition to part of the digital picture scaled in step (a).

8. (original) The method of claim 1 wherein step (b) is performed by using all of the data produced in step (a).

9. (currently amended) The method of claim 1 further comprising ~~removing~~ omitting part of the digital picture before performing steps (a) and (b) when a scaling factor is 2^k wherein k is a natural number.

10. (currently amended) A method for scaling a digital picture comprising the following steps:

(a) inputting a source picture to a source memory;

(b) providing a first buffer and a second buffer;

(c) determining scaling factors;

(d) generating initial data in the first buffer and second buffer;

(e) transferring a portion of data of the digital picture from the source memory to the first buffer, the portion of data having a size in a second direction of L and size in a first direction equal to a corresponding size of the source picture;

(f) using [[a]] an L-tap filter to scale the data in the first buffer in [[a]]] the first direction and storing the scaled data in the second buffer;

(g) using the L-tap filter to scale the data in the second buffer in [[a]] the second direction and storing the scaled data in a destination memory, the scaled data having a size in the first direction of 2*L-1; [[and]]

(h) for each different portion of data in sequence, repeating steps (e) through (g) to form a scaled picture; and

[[(h)]] (i) outputting [[a]] the scaled picture from the destination memory.

11. (cancelled)

12. (original) The method of claim 10 wherein the first direction is a horizontal direction, and the second direction is a vertical direction.

13. (original) The method of claim 10 wherein the first direction is a vertical direction, and the second direction is a horizontal direction.

14. (currently amended) The method of claim 10 wherein step (d) comprises ~~mirroring part of the digital picture~~ applying a mirror boundary condition to the first buffer and filtering initial data in the first buffer to produce the initial data in the second buffer.

15. (currently amended) The method of claim 10 further comprising before scaling a last portion of the digital picture in the second direction, ~~mirroring~~ applying a mirror boundary condition to part of the digital picture scaled in step (f).

16. (currently amended) The method of claim 10 further wherein step (e) comprises ~~removing~~ omitting transfer of part of the digital picture for down-scaling.